

# Simple and Fast Rounding Algorithms for Directed and Node-weighted Multiway Cut

Chandra Chekuri \*

Vivek Madan †

July 17, 2015

## Abstract

We study the multiway cut problem in *directed* graphs and one of its special cases, the *node-weighted* multiway cut problem in *undirected* graphs. In DIRECTED MULTIWAY CUT (DIR-MC) the input is an edge-weighted directed graph  $G = (V, E)$  and a set of  $k$  terminal nodes  $\{s_1, s_2, \dots, s_k\} \subseteq V$ ; the goal is to find a min-weight subset of edges whose removal ensures that there is no path from  $s_i$  to  $s_j$  for any  $i \neq j$ . In NODE-WEIGHTED MULTIWAY CUT (NODE-WT-MC) the input is a node-weighted undirected graph  $G$  and a set of  $k$  terminal nodes  $\{s_1, s_2, \dots, s_k\} \subseteq V$ ; the goal is to remove a min-weight subset of nodes to disconnect each pair of terminals. DIR-MC admits a 2-approximation [26] and NODE-WT-MC admits a  $2(1 - \frac{1}{k})$ -approximation [19], both via rounding of LP relaxations. Previous rounding algorithms for these problems, from nearly twenty years ago, are based on careful rounding of an *optimum* solution to an LP relaxation. This is particularly true for DIR-MC for which the rounding relies on a custom LP formulation instead of the natural distance based LP relaxation [26].

In this paper we describe extremely simple and near linear-time rounding algorithms for DIR-MC and NODE-WT-MC via a natural distance based LP relaxation. The dual of this relaxation is a special case of the maximum multicommodity flow problem. Our algorithms achieve the same bounds as before but have the significant advantage in that they can work with *any feasible* solution to the relaxation. Consequently, in addition to obtaining “book” proofs of LP rounding for these two basic problems, we also obtain significantly faster approximation algorithms by taking advantage of known algorithms for computing near-optimal solutions for maximum multicommodity flow problems. We also investigate lower bounds for DIR-MC when  $k = 2$  and in particular prove that the integrality gap of the LP relaxation is 2 even in directed planar graphs.

---

\*Dept. of Computer Science, University of Illinois, Urbana, IL 61801. Supported in part by NSF grant CCF-1319376. [chekuri@illinois.edu](mailto:chekuri@illinois.edu)

†Dept. of Computer Science, University of Illinois, Urbana, IL 61801. Supported in part by NSF grant CCF-1319376. [vmadan2@illinois.edu](mailto:vmadan2@illinois.edu)

# 1 Introduction

We study several variants of the multiway cut problem in graphs (also referred to as the multi-terminal cut problem). In the classical  $s$ - $t$  cut problem the input consists of a graph  $G = (V, E)$  and two distinct nodes  $s, t$ ; the goal is to separate  $s$  from  $t$  by removing a minimum cost set of edges and/or nodes. In the multiway cut problem the input is a graph  $G = (V, E)$  and a set  $S = \{s_1, s_2, \dots, s_k\}$  of  $k$  nodes from  $V$  called terminals; the goal is to separate the terminals from each other at minimum cost by removing edges and/or nodes. We describe the three main variants that are of interest to us.

**MULTIWAY CUT (EDGE-WT-MC):** The input is an undirected graph  $G = (V, E)$  along with non-negative edge weights  $w(e), e \in E$  and a set  $\{s_1, \dots, s_k\} \subseteq V$  of terminals. The goal is to find a min-cost set of edges  $E' \subseteq E$  such that in  $G - E'$  there is no path from  $s_i$  to  $s_j$  for  $i \neq j$ .

**NODE-WEIGHTED MULTIWAY CUT (NODE-WT-MC):** The input is an undirected graph  $G = (V, E)$  along with non-negative node weights  $w(v), v \in V$  and a set  $\{s_1, \dots, s_k\} \subseteq V$  of terminals. The goal is to find a min-cost set of nodes  $V' \subseteq V$  such that in  $G - V'$  there is no path from  $s_i$  to  $s_j$  for  $i \neq j$ .<sup>1</sup>

**DIRECTED MULTIWAY CUT (DIR-MC):** The input is a directed graph  $G = (V, E)$  along with non-negative edge weights  $w(e), e \in E$  and a set  $\{s_1, \dots, s_k\} \subseteq V$  of terminals. The goal is to find a min-cost set of edges  $E' \subseteq E$  such that in  $G - E'$  there is no path from  $s_i$  to  $s_j$  for  $i \neq j$ .

**Remark 1.1.** DIR-MC with  $k = 2$  is not the same as the  $s$ - $t$  cut problem. The goal is to separate  $s_1$  from  $s_2$  and  $s_2$  from  $s_1$ . In fact DIR-MC with  $k = 2$  is NP-Hard [17].

The complexity of the multiway cut problem and its variants have been extensively studied since the paper of Dahlhaus *et al.* [11]. They showed that EDGE-WT-MC with  $k = 3$  is NP-Hard; it was later observed that the problem is also APX-hard to approximate. This is in contrast to the case of  $k = 2$  which can be solved in polynomial-time in undirected graphs via a reduction to the  $s$ - $t$  minimum-cut problem.

EDGE-WT-MC reduces in an approximation preserving fashion to NODE-WT-MC which in turn reduces in an approximation preserving fashion to DIR-MC [19]; it is also easy to see that in the directed case, node-weighted and edge-weighted versions are equivalent. The current best approximation ratio for EDGE-WT-MC stands at 1.2965 due to Sharma and Vondrák [28]. For NODE-WT-MC a  $2(1 - 1/k)$  approximation is known from the work of Garg, Vazirani and Yannakakis [19], and for DIR-MC a 2 approximation is known from the work of Naor and Zosin [26]. VERTEX COVER reduces to NODE-WT-MC and DIR-MC in an approximation preserving fashion [19]. Assuming  $P \neq NP$  VERTEX COVER is hard to approximate to within a factor of 1.36 [12], and assuming the Unique Games Conjecture it is hard to approximate to within a factor of  $(2 - \varepsilon)$  for any fixed  $\varepsilon > 0$  [23]. These hardness results apply to NODE-WT-MC and DIR-MC and show that EDGE-WT-MC is provably easier to approximate than them.

Our focus in this paper is on approximation algorithms for NODE-WT-MC and DIR-MC. The known algorithms are based on rounding suitable LP relaxations for the problems. For both problems there is a simple and natural LP relaxation based on distance variables on nodes/edges; see Section 2 and 3. (We note that a similar relaxation applies to the more general MULTICUT problem and that dual of the LP relaxation corresponds to the LP for maximum multicommodity flow.) For NODE-WT-MC the algorithm of Garg, Vazirani and Yannakakis [19] shows that any *optimum* solution to the relaxation can be converted to a half-integral optimum solution which can then be rounded easily. The situation for DIR-MC is much more involved. Unlike the case of NODE-WT-MC, half-integral optimum solutions may not exist for the relaxation even for  $k = 2$ . Garg *et al.* [17] obtained an  $O(\log k)$ -approximation via the relaxation using ideas from approximation algorithms for multicut [18]. Naor and Zosin obtained a 2-approximation for DIR-MC in an elegant, surprising and somewhat

<sup>1</sup> In this definition terminals are allowed to be removed. If they are not allowed to be removed we can simply make their weight  $\infty$ .

mysterious fashion. They write a different LP relaxation called the *relaxed multiway flow* relaxation which is within a factor of 2 of the natural relaxation, and show that an *optimum* solution to this new relaxation can be rounded without any loss in the approximation. This gives an indirect proof that the natural relaxation has an integrality gap of at most 2. The proof of correctness crucially relies on complementary slackness properties of the optimum solution and is partly inspired by the ideas in [19]. The idea of using a relaxed multiway flow is inspired by earlier work on the subset feedback vertex problem [14].

The algorithms of [19] and [26] are from almost twenty years ago. During this intervening years no alternative algorithms or rounding schemes have been obtained for these basic problems. We observe that for the case of EDGE-WT-MC there is an extremely simple rounding scheme that converts any fractional feasible solution to a multiway cut with a loss of a factor of 2 (see [29]). The algorithm picks a random  $\theta \in (0, 1/2)$  and for each terminal  $s_i$  removes the edges leaving the ball  $B(s_i, \theta)$  of nodes contained within a radius  $\theta$  around  $s_i$  (with respect to distances given by the LP solution); more formally the output is  $\bigcup_{i=1}^k \delta(B(s_i, \theta))$ .

In this paper we show that very simple algorithms which are essentially similar in spirit to the above scheme also work for DIR-MC and NODE-WT-MC!

- The rounding algorithms are extremely simple and natural to describe, and in retrospect also to analyze.
- The algorithms only require a feasible solution to the natural LP relaxation and not necessarily an optimum solution.
- Given a feasible fractional solution, the rounding algorithms can be implemented in time that is similar to what is required for one single-source shortest path computation. The deterministic version requires an additional logarithmic factor.

In addition to algorithmic results we also obtain some lower bound results for DIR-MC with  $k = 2$ ; the goal is to separate  $s$  from  $t$  and  $t$  from  $s$  in a directed graph  $G$ ; subsequently we refer to this special case as *st*-Bi-Cut. We prove that the natural LP relaxation has an integrality gap of 2 for *st*-Bi-Cut even in *planar* directed graphs.

We believe that our algorithms and analysis will be useful for related problems. Indeed one of our motivations for simplifying the rounding schemes for DIR-MC and NODE-WT-MC came from attempts to obtain algorithms for a problem with applications to network information theory [8]. A significant consequence of our rounding algorithms are much faster approximation algorithms for NODE-WT-MC and DIR-MC in both theory and practice. Solving the LP relaxations for NODE-WT-MC and DIR-MC to optimality is quite challenging. The options are to use the Ellipsoid method or to use a compact formulation with a very large number of variables and constraints. As we remarked earlier, the dual of the natural LP relaxation for these problems is the maximum multicommodity flow problem. Combinatorial fully-polynomial time approximation schemes for solving these multicommodity flow problems have been extensively investigated in theoretical computer science and mathematical programming with a number of techniques developed over the years; we refer the reader to [27, 20, 30, 2, 16, 15, 3, 24]. Thus, a fast  $(1 + \varepsilon)$ -approximation for the LP relaxation for NODE-WT-MC and DIR-MC can be obtained using these methods. The fastest theoretical algorithms run in time  $\tilde{O}(m^2/\varepsilon^2)$  [15, 16] or in even faster  $\tilde{O}(mn/\varepsilon^2)$  time [24] under some mild conditions; here  $m$  is the number of edges and  $n$  is the number of nodes in  $G$  and  $\tilde{O}$  suppresses poly-logarithmic factors. Note that these running times are independent of  $k$ . Our rounding algorithms can convert such an approximate feasible solution to an integral cut in near-linear time with a factor of 2 loss in the cost. Thus, we can obtain provably fast  $(2 + \varepsilon)$ -approximation algorithms. Since our focus is on the rounding algorithms we do not go into further details of specific algorithms or running times for solving the relaxation.

We refer the interested reader to quickly jump to Section 2 to see the simplicity of the rounding scheme and its analysis for DIR-MC that achieves a bound of 2. This also applies to NODE-WT-MC via a simple reduction to DIR-MC. We also discuss some new observations on the hardness of the problem when  $k = 2$ . In

Section 3 we give a slightly different rounding scheme for NODE-WT-MC that achieves an improved bound of  $2(1 - 1/k)$ , matching the known ratio from [19].

## 1.1 Other related work

The natural LP relaxation for EDGE-WT-MC has an integrality gap of  $2(1 - 1/k)$ . Approximation algorithms for EDGE-WT-MC received substantial attention following the breakthrough work of Calinescu, Karloff and Rabani [5]. They developed a new “geometric” LP relaxation (henceforth referred to as the CKR-relaxation) which they used to obtain a  $(1.5 - 1/k)$ -approximation. The integrality gap of the CKR-relaxation, and consequently the approximation ratio, was improved subsequently to 1.3438 by Karger *et al.* [22], to 1.32388 by Buchbinder *et al.* [4], and to the currently best known bound of 1.2965 by Sharma and Vondrák [28]. For  $k = 3$  a tight bound of 12/11 is known [9, 22]. It is also known that assuming the Unique Games Conjecture, for any fixed  $k$ , the approximability threshold for EDGE-WT-MC coincides with the integrality gap of the CKR-relaxation [25].

The CKR-relaxation makes use of the observation that EDGE-WT-MC can be viewed as a partition problem where the goal is to partition the node set  $V(G)$  into  $k$  parts  $V_1, \dots, V_k$  to minimize  $\sum_{i=1}^k w(\delta(V_i))$  subject to the constraint that for  $1 \leq i \leq k$ ,  $s_i \in V_i$ . SUBMODULAR MULTIWAY PARTITION (SUB-MP) is a generalization from the setting of graphs to arbitrary submodular functions. Here we are given a non-negative submodular function  $f : 2^V \rightarrow \mathbb{R}^+$  over the ground set  $V$  along with terminals  $\{s_1, \dots, s_k\} \subset V$ . The goal is to partition  $V$  into  $V_1, \dots, V_k$  to minimize  $\sum_{i=1}^k f(V_i)$  subject to the constraint that  $s_i \in V_i$  for  $1 \leq i \leq k$ . If  $f$  is symmetric, as in the case of the undirected graph cut function, we obtain the SYMMETRIC SUBMODULAR MULTIWAY PARTITION (SYM-SUB-MP) problem. These problems were considered by Zhao, Nagamochi and Ibaraki [31] who analyzed greedy-splitting algorithms, and more recently by Chekuri and Ene [6] who used a Lovász-extension based convex relaxation. Interestingly, the convex relaxation when specialized to EDGE-WT-MC yields the CKR-relaxation. Chekuri and Ene [6] obtained a  $(1.5 - 1/k)$ -approximation for SYM-SUB-MP and 2-approximation for SUB-MP. Ene, Vondrák and Wu [13] improved the bound for SUB-MP to  $2(1 - 1/k)$  and also obtained lower bound results in the oracle model.

NODE-WT-MC cannot be viewed as a partition problem directly. Nevertheless, it can be seen that NODE-WT-MC is equivalent to HYPERGRAPH MULTIWAY CUT problem (HYPERGRAPH-MC) which is a generalization of EDGE-WT-MC from graphs to hypergraphs. HYPERGRAPH-MC can be cast as a special case of SUB-MP (note that the reduction uses a non-symmetric submodular function  $f$ ) and thus NODE-WT-MC can be indirectly reduced to a partition problem. This leads to an alternative  $2(1 - 1/k)$ -approximation for NODE-WT-MC based on the Lovász-extension based relaxation for HYPERGRAPH-MC. This relaxation does not result in a better worst-case approximation than the distance-based relaxation, however, it appears to be strictly stronger in that it improves the approximation ratio in special some cases as observed in [7]. No fast approximation algorithms are known to solve this convex relaxation.

Finally we mention the MULTICUT problem where the goal is to separate a given set of  $k$  node-pairs  $(s_1, t_1), \dots, (s_k, t_k)$  in a given graph at minimum-cost. One can consider undirected graphs with edge weights, undirected graphs with node weights and directed graph with edge weights. These versions generalize the corresponding multiway cut problems. The best known approximation ratio for MULTICUT in undirected graphs is  $O(\log k)$  [18, 17] while the best known bounds in directed graphs is  $\min(k, \tilde{O}(n^{11/23}))$  [1]. Moreover, it is known from the work of Chuzhoy and Khanna [10] that the problem in directed graphs is inapproximable to a factor better than  $\tilde{\Omega}(2^{\log^{1-\varepsilon} n})$ .

## 2 LP Relaxation and rounding for DIR-MC

DIR-MC can be naturally formulated as an integer linear program with variables  $x_e \in \{0, 1\}$ ,  $e \in E$  which indicate whether  $e$  is cut or not. Let  $\mathcal{P}_{ij}$  be the set of all directed paths from  $s_i$  to  $s_j$  in  $G$ . The constraint that  $s_i$  is separated from  $s_j$  by the cut can be enforced by requiring that  $\sum_{e \in p} x_e \geq 1$  for each  $p \in \mathcal{P}_{ij}$ . This leads to the following LP relaxation where the integer constraint  $x_e \in \{0, 1\}$  is replaced by  $x_e \in [0, 1]$ . We can without loss of generality drop the constraint  $x_e \leq 1$ .

**DIR-MC-REL**

$$\begin{aligned} \min \quad & \sum_{e \in E} w_e x_e \\ & \sum_{e \in p} x_e \geq 1 \quad p \in \mathcal{P}_{ij}, i \neq j \\ & x_e \geq 0 \quad e \in E \end{aligned}$$

Figure 1: LP Relaxation for DIR-MC

The main result of the paper is the following theorem.

**Theorem 2.1.** *There is a randomized algorithm that given a feasible solution  $\mathbf{x}$  to DIR-MC-REL returns a feasible integral solution of expected cost at most  $2 \sum_e w_e x_e$ , and runs in  $O(m + n \log n)$  time. The algorithm can be derandomized to yield a deterministic 2-approximation algorithm that runs in  $O(m \log n)$  time. Here,  $m = |E(G)|$ ,  $n = |V(G)|$ .*

We now describe the simple randomized ball-cutting algorithm that achieves the properties claimed by the theorem. Let  $\mathbf{x}$  be a feasible solution to DIR-MC-REL. For any two nodes  $u, v \in V$  we define  $d_x(u, v)$  be the shortest path length from  $u$  to  $v$  using edge lengths given by  $\mathbf{x}$ . For notational simplicity we omit the subscript  $x$  since there is little chance of confusion. The algorithm adds new nodes  $t_1, t_2, \dots, t_k$  and adds the edge set  $\{(t_i, s_j) \mid i \neq j\}$  and sets the  $x$  value of each of these new edges to 0. Note that, this is in effect a reduction of the DIR-MC for the given instance to a DIR-MULTICUT instance which requires us to separate the pairs  $(t_i, s_i)$ ,  $1 \leq i \leq k$ . The solution  $\mathbf{x}$  augmented with the extra nodes and edges leads to a feasible fractional solution for this DIR-MULTICUT instance. Our algorithm, formally described below, is very simple. We pick a random  $\theta \in (0, 1)$  and take the union of the cuts defined by balls of radius  $\theta$  around each  $t_i$ . More formally let  $B(v, r)$  be the set of all nodes at distance at most  $r$  from  $v$ . Then the algorithm simply outputs  $\bigcup_{i=1}^k \delta^+(B(t_i, \theta))$  where  $\delta^+(A)$  denote the set of outgoing edges from  $A$ .

---

### Algorithm 1 Rounding for DIR-MC

---

- 1: Given a feasible solution  $\mathbf{x}$  to DIR-MC-REL
  - 2: Add new vertices  $t_1, \dots, t_k$ , edges  $(t_i, s_j)$  for all  $i \neq j$  and set  $x(t_i, s_j) = 0$
  - 3: Pick  $\theta \in (0, 1)$  uniformly at random
  - 4:  $C = \bigcup_{i=1}^k \delta^+(B(t_i, \theta))$
  - 5: Return  $C$
- 

Note that  $C$  is a random set of edges that depends on the choice of  $\theta$ . We denote by  $C(\theta)$  the set of edges output by the algorithm for a given  $\theta$ .

**Lemma 2.2.** *If  $\mathbf{x}$  is a feasible fractional solution to DIR-MC-REL,  $C(\theta)$  is a feasible multiway cut for  $\{s_1, \dots, s_k\}$  for any  $\theta \in (0, 1)$ . Thus, Algorithm 1 always returns a feasible integral solution given a feasible  $\mathbf{x}$ .*

**Proof:** Fix any  $i \in \{1, \dots, k\}$  and  $\theta \in (0, 1)$ . Since  $d(t_i, s_j) = 0$  for all  $j \neq i$ , we have that  $s_j \in B(t_i, \theta)$  for all  $j \neq i$ . Moreover, by feasibility of  $\mathbf{x}$ , we have  $d(t_i, s_i) \geq 1$  for otherwise there will be a path of length less than 1 from some  $s_j$  to  $s_i$  where  $j \neq i$ . Therefore  $s_i \notin B(t_i, \theta)$  because  $\theta < 1$ . Therefore,  $G - \delta^+(B(t_i, \theta))$  has no path from  $s_j$  to  $s_i$  for any  $j \neq i$ . Since  $C(\theta) = \bigcup_i \delta^+(B(t_i, \theta))$ , it follows that there is no path in  $G - C(\theta)$  from  $s_j$  to  $s_i$  for any  $j \neq i$ .  $\square$

We now bound the probability that any fixed edge  $e$  is cut by the algorithm, that is,  $\Pr[e \in C]$ . Note that  $e$  may be *simultaneously* cut by several  $t_i$  for the same value of  $\theta$  but we are only interested in the probability that it is included in  $C$ .

**Lemma 2.3.** *For any edge  $e \in E$ ,  $\Pr[e \in C] \leq 2x_e$ .*

**Proof:** Let  $e = (u, v)$ . Rename the terminals such that  $d(s_1, u) \leq d(s_2, u) \leq \dots \leq d(s_k, u)$ . This implies that

$$d(t_1, u) = d(s_2, u)$$

and

$$d(t_2, u) = d(t_3, u) = \dots = d(t_k, u) = d(s_1, u).$$

Edge  $e \in \delta^+(B(t_i, \theta))$  if and only if  $\theta \in [d(t_i, u), d(t_i, v)]$ ; we have that  $d(t_i, v) \leq d(t_i, u) + x_e$ . Defining the interval  $I_i$  as  $[d(t_i, u), d(t_i, u) + x_e]$ , we see that  $e \in \delta^+(B(t_i, \theta))$  only if  $\theta \in I_i$ . However, from the property that  $d(t_2, u) = d(t_3, u) = \dots = d(t_k, u)$ ,  $I_2 = I_3 = \dots = I_k$ . Thus,  $e \in C$  only if  $\theta \in I_1$  or  $\theta \in I_2$  and since  $|I_1|$  and  $|I_2|$  are both at most  $x_e$  long and  $\theta$  is chosen uniformly at random from  $(0, 1)$ ,

$$\Pr[e \in C] \leq \Pr[\theta \in I_1] + \Pr[\theta \in I_2] \leq 2x_e.$$

$\square$

**Corollary 2.4.**  $\mathbb{E}[C]$ , the expected cost of  $C$ , is at most  $2 \sum_e w_e x_e$ .

**Running time analysis and derandomization:** A natural implementation of Algorithm 1 would first choose  $\theta$  and then compute  $\delta^+(B(t_i, \theta))$  for each  $i$ . This can be easily accomplished via  $k$  executions of Dijkstra's single-source shortest path algorithm, one for each  $t_i$ , leading to a running time of  $O(k(m + n \log n))$  where  $m = |E|$  and  $n = |V|$ . However, by taking advantage of our analysis in Lemma 2.3, we can obtain a run time that is equivalent to a single execution of Dijkstra's algorithm.

Consider a slight variation of Algorithm 1. For each edge  $e = (u, v)$ , define two intervals  $I_1(e) = [d(s_1, u), d(s_1, u) + x_e]$  and  $I_2(e) = [d(s_1, u), d(s_1, u) + x_e]$ , where  $s_1, s_2$  are the two terminals from which  $u$  is the closest in terms of distance. We pick  $\theta \in (0, 1)$  uniformly at random and include  $e$  in  $C$  iff  $\theta \in I_1(e)$  or  $\theta \in I_2(e)$ . The analysis in Lemmas 2.2 and 2.3 shows that even this modified algorithm outputs a feasible cut whose expected cost is at most  $2 \sum_e w_e x_e$ . Note that the edges cut by this modified algorithm may be a strict superset of the edges cut by Algorithm 1. The advantage of the modified algorithm is that we only need to calculate  $I_1(e)$  and  $I_2(e)$  for each edge  $e \in E$ . To do this, for each node  $u$ , we need to find the two terminals from which  $u$  is the closest and their corresponding distances. More formally, consider the following  $h$ -nearest-terminal problem.

**Problem 1.** *Given a directed graph  $G = (V, E)$  with non-negative edge-lengths, a set  $S \subseteq V(G)$  of  $k$  terminals, and an integer  $h \leq k$ , for each vertex  $v$ , find the  $h$  terminals from which  $v$  is the closest among the terminals and their corresponding distances. In other words for each  $v$  find the  $h$  smallest values in  $d(s_1, v), d(s_2, v), \dots, d(s_k, v)$  where  $S = \{s_1, \dots, s_k\}$ .*

The above problem can be solved via a randomized algorithm using hashing that runs in expected time  $O(h(m + n \log n))$ , which corresponds to  $h$  executions of Dijkstra's algorithm. It can also be solved in

$O(hm \log h + hn \log n)$  time via a deterministic algorithm. See [21] who refers to this as the  $h$ -nearest-neighbors problem.

Using the algorithm for the  $h$ -nearest-terminal problem with  $h = 2$ , we can calculate  $I_1(e)$  and  $I_2(e)$  for each  $e \in E$  in  $O(m + n \log n)$  time<sup>2</sup>. We then chose  $\theta$  uniformly at random from  $(0, 1)$  and cut  $e$  if  $\theta$  lies in one of the range  $I_1(e)$  or  $I_2(e)$ . This gives us a 2-approximate randomized algorithm with running time  $O(m + n \log n)$ .

We can derandomize the algorithm by computing the cheapest cut among all  $\theta \in (0, 1)$  as follows. Once  $I_1(e)$  and  $I_2(e)$  are computed for each  $e$  we sort the  $4m$  end points of these  $2m$  intervals; let them be  $\theta_1 \leq \theta_2 \leq \dots \leq \theta_{4m}$ . We observe that it suffices to evaluate the cut value at each of these values of  $\theta$ . A simple scan of these  $4m$  points while updating the cut-value at each end point can be accomplished in  $O(m)$  time. Sorting the end points takes  $O(m \log n)$  time. This leads to a deterministic 2-approximation algorithm with running time  $O(m \log n)$ .

## 2.1 DIR-MC with $k = 2$

In this section we address DIR-MC with  $k = 2$  which we refer to as  $st$ -Bi-Cut. We believe this is an interesting problem on its own as it is related closely to the classical  $s$ - $t$  cut problem. As we remarked earlier,  $st$ -Bi-Cut is NP-Hard and APX-Hard to approximate. This was shown in [17, 19] via a simple approximation preserving reduction from EDGE-WT-MC with  $k = 3$ . Another consequence of the reduction is that the integrality gap of DIR-MC-REL for  $st$ -Bi-Cut is at least  $4/3$ . On the other hand no ratio better than 2 is known for  $st$ -Bi-Cut. This naturally raises the following question.

**Question 1.** *What is the integrality gap of DIR-MC-REL for  $st$ -Bi-Cut? What is the approximability of  $st$ -Bi-Cut?*

We obtain two theorems. The first one shows that the integrality gap for  $st$ -Bi-Cut is 2.

**Theorem 2.5.** *Integrality gap of DIR-MC-REL for  $st$ -Bi-Cut is 2 even in planar directed graphs.*

The second theorem slightly extends a result in [19].

**Theorem 2.6.** *There is an approximation preserving reduction from 4-terminal NODE-WT-MC to  $st$ -Bi-Cut.*

We raise the following question.

**Question 2.** *Can we prove a factor 2 hardness of approximation for DIR-MC under the assumption that  $P \neq NP$ ? Does a factor of 2 hardness hold for  $st$ -Bi-Cut even under the Unique Games conjecture?*

**Integrality gap construction:** Proof of Theorem 2.5 is based on recursively defined sequence of graphs  $G_0, G_1, \dots, G_h$  with increasing integrality gap; we will use  $\alpha_i$  to denote the integrality gap (we also refer to this as the flow-cut gap) in  $G_i$ . The two terminals will be denoted by  $s, t$ . The symmetry in the construction will ensure that in  $G_i$  the  $s$ - $t$  cut value will be equal to the  $t$ - $s$  cut value; we refer to these common values as the one-way cut value and the optimum value of a cut that separates  $s$  from  $t$  and  $t$  from  $s$  as the two-way cut value. The graph  $G_0$  is shown in Fig 2 and it is easy to see that  $\alpha_0 = 1$ .

The iterative construction of  $G_{i+1}$  from  $G_i$  is shown at a high-level in figure 2. A formal description is as follows. To obtain  $G_{i+1}$  with terminals  $s, t$  we start with two copies of  $G_i$  with terminals  $s_1, t_1$  and  $s_2, t_2$  (denoted by  $H, H'$ ) and two new vertices  $v_1, v_2$ . We set  $s = s_1, t = t_2$  and identify  $t_1$  and  $s_1$  as the

---

<sup>2</sup>One can easily derive the  $h = 2$  case from first principles also.

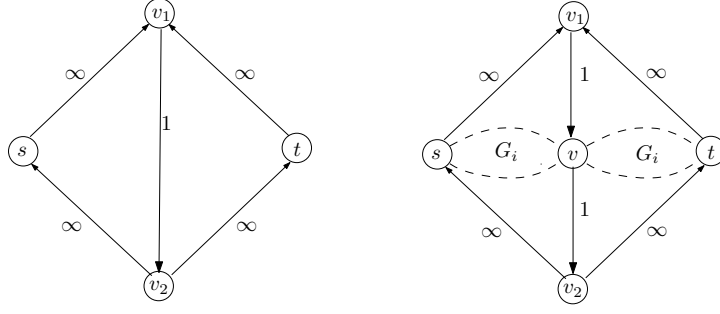


Figure 2:  $G_0$  on the left and constructing  $G_{i+1}$  from  $G_i$  shown on the right.

center vertex  $v$  shown in the figure. We add edges  $(v_1, v)$  and  $(v, v_2)$  with weight 1 and four other edges  $\{(s, v_1), (t, v_1), (v_2, s), (v_2, t)\}$  each with weight infinity. Finally we scale the weights of the edges of  $H$  and  $H'$  such that the two-way cut value in each of them is  $\frac{\alpha_i}{2-\alpha_i}$ . It is easy to observe inductively that the each graph in the sequence is planar and moreover the graph can be embedded such that  $s$  and  $t$  are on the outer face. The analysis of the integrality gap of this construction can be found in the appendix.

Subsequent to our construction, Julia Chuzhoy obtained an alternative non-recursive construction with an integrality gap of 2 for  $st$ -Bi-Cut.

**Reduction from 4-terminal NODE-WT-MC to  $st$ -Bi-Cut:** Given a NODE-WT-MC instance with graph  $G$  and set of terminals  $\{s_1, s_2, s_3, s_4\}$ , Figure 3 shows the ingredients of a reduction to DIR-MC instance with graph  $G'$  and terminals  $s, t$ . This is a slight modification of the reduction from three-terminal EDGE-WT-MC to  $st$ -Bi-Cut given in [19]. It is convenient to consider the node-weighted version of DIR-MC which is equivalent to the edge-weighted version. Formally  $G'$  is obtained from  $G$  by the addition of two new nodes  $s, t$  which are connected to the terminals via directed edges of infinite weight as shown in the figure. Each edge  $uv \in E(G)$  is replaced by two directed edges  $(u, v)$  and  $(v, u)$  and the weights of the nodes of  $G$  remain the same. We will assume without loss of generality that the terminals  $s_1, s_2, s_3, s_4$  have infinite weight. A relatively simple case analysis shows that  $C \subset V(G)$  is a feasible node-multiway cut for the terminals  $\{s_1, \dots, s_4\}$  in  $G$  iff  $C$  is a feasible node-multiway cut in  $G'$  for  $\{s, t\}$ . This type of reduction does not seem to generalize beyond four terminals.

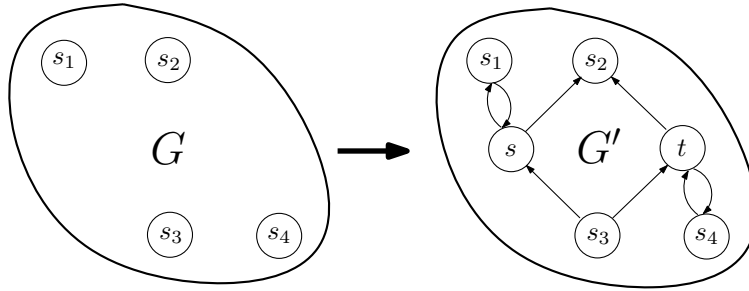


Figure 3: Reduction from 4-terminal NODE-WT-MC to  $st$ -Bi-Cut. Non-terminal vertices are not shown.

Garg *et al.* [19] showed that DIR-MC-REL does not necessarily have half-integral optimum solutions. In Section B we extend their example to show that for every non-negative integer  $\ell$  there exist instances for which there is no optimum solution to DIR-MC-REL that is  $1/\ell$  integral.



### 3 LP Relaxation and rounding for NODE-WT-MC

The LP relaxation for the NODE-WT-MC is similar to the one for EDGE-WT-MC. We have a variable  $x_v \in \{0, 1\}$  for each  $v \in V$  which indicates whether to remove  $v$  or not. We can assume without loss of generality that we cannot remove the terminals  $s_1, s_2, \dots, s_k$  and moreover that they form an independent set. This can be accomplished by adding to each original terminal  $s_i$  a new dummy terminal  $s'_i$  and adding the edge  $s_i s'_i$ . Let  $\mathcal{P}_{ij}$  be the set of all paths between  $s_i$  and  $s_j$  in  $G$ . Note that in the undirected graph case we do not need to distinguish  $P_{ij}$  from  $P_{ji}$ . Let  $S = \{s_1, s_2, \dots, s_k\}$  be the set of terminals.

NODE-MC-REL			
min	$\sum_{v \in V \setminus S} w_v x_v$		
	$\sum_{v \in p} x_v$	$\geq 1$	$p \in \mathcal{P}_{ij}, i < j$
	$x_v$	$= 0$	$v \in S$
	$x_v$	$\geq 0$	$v \in V$

Figure 4: LP Relaxation for NODE-WT-MC

**Theorem 3.1.** *There is a polynomial-time randomized algorithm that given a feasible solution  $\mathbf{x}$  to NODE-MC-REL returns a feasible integral solution of expected cost at most  $2(1 - 1/k) \sum_v w_v x_v$ , and runs in  $O(m + n \log n)$  time. The algorithm can be derandomized to yield a deterministic 2-approximation algorithm that runs in  $O(kn + m + n \log n)$  time.*

Let  $\mathbf{x}$  be a feasible fractional solution to NODE-MC-REL. For nodes  $u$  and  $v$  we define  $d_x(u, v)$  to be the length of the shortest path between  $u$  and  $v$  according to the node weights given by  $\mathbf{x}$ ; we count the weights of the end points  $u$  and  $v$  in  $d_x(u, v)$ . We omit the subscript  $x$  in subsequent discussion. For a given radius  $r$  and node  $u$  let  $B(u, r)$  be the set of all nodes  $v$  such that  $d(u, v) \leq r$ ;  $B(u, r)$  is the ball of radius  $r$  around  $u$ . We define the “boundary” of radius  $r$  from  $u$ , denoted by  $B^+(u, r)$  to be the set of all nodes that are not in  $B(u, r)$  but have an edge to some node in  $B(u, r)$ .

**Proposition 3.2.** *A node  $v \in B^+(u, r)$  iff  $r < d(u, v) \leq r + x_v$ . Further, if  $v \in B^+(u, r)$  for  $r < 1$  then  $x_v \neq 0$ .*

Our rounding algorithm first picks an index  $\ell$  uniformly at random from  $\{1, 2, \dots, k\}$ . It then picks a  $\theta$  uniformly at random from  $(0, 1/2)$ . For each  $i \neq \ell$  it includes in the final cut  $C$  all nodes  $v$  that are in the “boundary” of the ball of radius  $\theta$  around  $s_i$ . The formal description is given in Algorithm 2.

---

#### Algorithm 2 Rounding for NODE-WT-MC

---

- 1: Given feasible fractional solution  $\mathbf{x}$  to NODE-MC-REL
  - 2: Chose  $\ell \in \{1, 2, \dots, k\}$  uniformly at random
  - 3: Pick  $\theta \in (0, 1/2)$  uniformly at random
  - 4:  $C = \cup_{i \neq \ell} B^+(s_i, \theta)$
  - 5: Return  $C$
- 

Let  $C(\ell, \theta)$  be the output of the algorithm for fixed  $\ell$  and  $\theta$ . We first argue that the algorithm always returns a feasible multiway cut.

**Lemma 3.3.** For all  $\ell, \theta$ ,  $C(\ell, \theta)$  is a feasible multiway cut for the given instance. That is,  $G - C(\ell, \theta)$  has no path from  $s_i$  to  $s_j$  for  $i \neq j$ .

**Proof:** Consider any pair  $i, j \in \{1, 2, \dots, k\}$  where  $i \neq j$ . Assume  $i \neq \ell$ , the case when  $j \neq \ell$  is similar. The ball  $B(s_i, \theta)$  does not contain  $s_j$  since  $\theta < 1/2$  and  $d(s_i, s_j) \geq 1$  by feasibility of  $\mathbf{x}$ .  $C(\ell, \theta)$  contains all nodes from  $B^+(s_i, \theta)$ , thus, in  $G - C(\ell, \theta)$  there cannot be a path from  $s_i$  to any node in  $V \setminus B(s_i, \theta)$ , and hence to  $s_j$ .  $\square$

We say that  $v$  is cut by the algorithm if  $v \in C$ . The key to the performance guarantee of the algorithm is the following lemma.

**Lemma 3.4.**  $\Pr[v \in C] \leq 2(1 - 1/k)x_v$ .

**Proof:** Fix a node  $v$  and rename the terminals such that  $d(s_1, v) \leq d(s_2, v) \leq \dots \leq d(s_k, v)$ . Define the interval  $I_i$  as  $[d(s_i, v) - x_v, \min(d(s_i, v), 1/2))$ . From the algorithm description and Proposition 3.2, we can see that  $v \in C$  iff  $\exists i$  such that  $\ell \neq i$  and  $\theta \in I_i$ .

Note that  $I_i$  is an empty interval if  $x_v = 0$  or  $d(s_1, v) - x_v \geq 1/2$ . Hence we can assume that  $x_v > 0$  and  $d(s_1, v) - x_v < 1/2$ , otherwise  $I_i$  is empty for all  $i$  and  $\Pr[v \in C] = 0$ . We now consider two cases depending on whether  $d(s_2, v) - x_v$  is greater than  $1/2$  or not.

First, consider the case when  $d(s_2, v) - x_v \geq 1/2$ . Interval  $I_2$  is empty. Since  $d(s_2, v) \leq d(s_3, v) \leq \dots \leq d(s_k, v)$ , intervals  $I_3, I_4, \dots, I_k$  are also empty. Hence,  $v \in C$  iff  $\ell \neq 1$  and  $\theta \in I_1$ . Interval  $I_1$  has length at most  $x_v$  and  $\theta$  is chosen uniformly at random from  $(0, 1/2)$ . Therefore,

$$\Pr[v \in C] = \Pr[\ell \neq 1] \Pr[\theta \in I_1] \leq \left(1 - \frac{1}{k}\right) \cdot 2x_v.$$

In the preceding equation we used independence in the choice of  $\ell$  and  $\theta$ .

Next, consider the case when  $d(s_2, v) - x_v < 1/2$ . From the feasibility of  $\mathbf{x}$ , we have that  $d(s_1, v) - x_v + d(s_2, v) \geq 1$  (recall that  $d(s_1, v)$  and  $d(s_2, v)$  include the length of  $x_v$ ). This implies that  $d(s_1, v) \geq 1/2$ . Since,  $d(s_i, v) \geq d(s_1, v)$  for all  $i$ , we have  $d(s_i, v) \geq 1/2$  which implies that for all  $i$ ,  $I_i = [d(s_i, v) - x_v, 1/2)$ . Easy to see that  $I_1 \supseteq I_2 \supseteq \dots \supseteq I_k$ . Therefore,  $v \in C$  iff  $\ell = 1$  and  $\theta \in I_2$  or  $\ell \neq 1$  and  $\theta \in I_1$ . Length of interval  $I_1$  and  $I_2$  are  $1/2 - d(s_1, v) + x_v$  and  $1/2 - d(s_2, v) + x_v$  respectively. Hence,

$$\begin{aligned} \Pr[v \in C] &= \Pr[\ell = 1] \Pr[\theta \in I_2] + \Pr[\ell \neq 1] \Pr[\theta \in I_1] \\ &= 1/k \cdot 2(1/2 - d(s_2, v) + x_v) + (1 - 1/k) \cdot 2(1/2 - d(s_1, v) + x_v) \\ &\leq 2(1 - 1/k)(1 - d(s_1, v) - d(s_2, v) + 2x_v) \\ &\leq 2(1 - 1/k)x_v \end{aligned}$$

In the penultimate inequality above, we use the fact that  $1 - 1/k \geq 1/k$  if  $k \geq 2$ . The final inequality follows from already stated observation,  $d(s_1, v) + d(s_2, v) - x_v \geq 1$  due to feasibility of  $\mathbf{x}$ .  $\square$

**Corollary 3.5.**  $\mathbb{E}[w(C)] = \sum_{v \in V} w_v \Pr[v \in C] \leq 2(1 - 1/k) \sum_v w_v x_v$ . Thus, the expected cost of the cut output by the algorithm is at most  $2(1 - 1/k)$  times the cost of the fractional solution  $\mathbf{x}$ .

**Running time:** Algorithm 2 can be implemented in  $O(m + n \log n)$  time, in a fashion very similar to the implementation of the modified version of Algorithm 1. First, we pick  $\ell$  uniformly at random from  $\{1, \dots, k\}$  and  $\theta$  uniformly at random from  $(0, 1/2)$ . Then, for each vertex  $v$  we find the closest terminal  $s$  in the set  $S \setminus \{s_\ell\}$  and cut vertex  $v$  if  $d(s, v) - x_v \leq \theta < d(s, v)$ . Finding nearest terminal for each vertex can be done in  $O(m + n \log n)$  time. Hence, we get a randomized  $2(1 - 1/k)$ -approximation rounding scheme in time  $O(m + n \log n)$ .

To derandomize, we consider for each  $v$  intervals  $I_1(v)$  and  $I_2(v)$  as in the proof of Lemma 3.4. Using the  $h$ -nearest terminal algorithm for  $h = 2$  with  $S$  as the set of terminals, in  $O(m + n \log n)$  time, we can compute

$I_1(v)$  and  $I_2(v)$  for all  $v$ . We sort the  $4n$  end points of these  $2n$  intervals and let them be  $\theta_1, \theta_2, \dots, \theta_{4n}$ . It suffices to find the cost of the cut for each  $\theta$  from this  $4n$  values and for each  $\ell \in \{1, 2, \dots, k\}$ . We process these sorted values in order and for each  $\theta$ , we calculate  $w(C(\ell, \theta))$  for all  $\ell$ . The proof of Lemma 3.4 shows that this can be done by using only  $I_1(v)$  and  $I_2(v)$  for all  $v$ . As we process the end points in the sorted order the time to update the cut for each  $\ell$  per end point is  $O(1)$ . Thus, in  $O(nk + m + n \log n)$  time we can obtain a deterministic algorithm that gives a  $2(1 - 1/k)$ -approximation.

**Acknowledgments:** CC thanks Sudeep Kamath, Sreeram Kannan and Pramod Viswanath for extensive discussions on the problems considered in [8] which inspired us to revisit the rounding schemes for multiway cut problems. CC also thanks Anupam Gupta for discussion on some of the problems considered in [8] during an Oberwolfach workshop.

## References

- [1] Amit Agarwal, Noga Alon, and Moses S Charikar. Improved approximation for directed cut problems. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 671–680. ACM, 2007.
- [2] Daniel Bienstock. *Potential Function Methods for Approximately Solving Linear Programming Problems: Theory and Practice: Theory and Practice*, volume 53. Springer Science & Business Media, 2002.
- [3] Daniel Bienstock and Garud Iyengar. Approximating fractional packings and coverings in  $o(1/\epsilon)$  iterations. *SIAM Journal on Computing*, 35(4):825–854, 2006.
- [4] Niv Buchbinder, Joseph Seffi Naor, and Roy Schwartz. Simplex partitioning via exponential clocks and the multiway cut problem. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 535–544. ACM, 2013.
- [5] Gruia Calinescu, Howard J. Karloff, and Yuval Rabani. An improved approximation algorithm for multiway cut. *Journal of Computer and System Sciences*, 60(3):564–574, 2000. Preliminary version in STOC 1998.
- [6] Chandra Chekuri and Alina Ene. Approximation algorithms for submodular multiway partition. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 807–816. IEEE, 2011.
- [7] Chandra Chekuri and Alina Ene. Submodular cost allocation problem and applications. In *International Colloquium on Automata, Languages and Programming (ICALP (I))*, pages 354–366, 2011. A longer version is available on the arXiv, abs/1105.2040.
- [8] Chandra Chekuri, Sudeep Kamath, Sreeram Kannan, and Pramod Viswanath. Delay-constrained unicast and the triangle-cast problem. In *Proc. of ISIT*, 2015.
- [9] Kevin KH Cheung, William H Cunningham, and Lawrence Tang. Optimal 3-terminal cuts and linear programming. *Mathematical Programming*, 106(1):1–23, 2006.
- [10] Julia Chuzhoy and Sanjeev Khanna. Polynomial flow-cut gaps and hardness of directed cut problems. *Journal of the ACM (JACM)*, 56(2):6, 2009.

- [11] Elias Dahlhaus, David S. Johnson, Christos H. Papadimitriou, Paul D. Seymour, and Mihalis Yannakakis. The complexity of multiterminal cuts. *SIAM Journal on Computing*, 23(4):864–894, 1994. Preliminary version in STOC 1992.
- [12] Irit Dinur and Shmuel Safra. On the hardness of approximating minimum vertex cover. *Annals of Mathematics*, 162(1):439–485, 2005. Preliminary version in *Proc. of ACM STOC*, 2002.
- [13] Alina Ene, Jan Vondrák, and Yi Wu. Local distribution and the symmetry gap: Approximability of multiway partitioning problems. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 306–325. SIAM, 2013.
- [14] Guy Even, Joseph Naor, and Leonid Zosin. An 8-approximation algorithm for the subset feedback vertex set problem. *SIAM Journal on Computing*, 30(4):1231–1252, 2000. Preliminary version in *Proc. of IEEE FOCS*, 1996.
- [15] Lisa K. Fleischer. Approximating fractional multicommodity flow independent of the number of commodities. *SIAM Journal of Discrete Mathematics*, 13(4):505–520, 2000. Preliminary version in FOCS’99.
- [16] Naveen Garg and Jochen Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. *SIAM J. on Computing*, 37:630–652, 2007.
- [17] Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. Multiway cuts in directed and node weighted graphs. In Serge Abiteboul and Eli Shamir, editors, *ICALP*, volume 820 of *Lecture Notes in Computer Science*, pages 487–498. Springer, 1994.
- [18] Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. Approximate max-flow min-(multi) cut theorems and their applications. *SIAM Journal on Computing*, 25(2):235–251, 1996.
- [19] Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. Multiway cuts in node weighted graphs. *Journal of Algorithms*, 50(1):49–61, 2004. Preliminary version in *Proc. of ICALP*, 1994.
- [20] M. Grigoriadis and L. Khachiyan. Fast approximation schemes for convex programs with many blocks and coupling constraints. *SIAM Journal on Optimization*, 4(1):86–107, 1994.
- [21] Sarel Har-Peled. Computing the  $k$ -nearest-neighbors for all vertices via dijkstra. April 2015. [sarielhp.org/p/14/k\\_nn/k\\_nn.pdf](http://sarielhp.org/p/14/k_nn/k_nn.pdf).
- [22] David R. Karger, Philip N. Klein, Clifford Stein, Mikkell Thorup, and Neal E. Young. Rounding algorithms for a geometric embedding of minimum multiway cut. *Mathematics of Operations Research*, 29(3):436–461, 2004. Preliminary version in STOC 1999.
- [23] Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within  $2 - \epsilon$ . *Journal of Computer and System Sciences*, 74(3):335–349, 2008.
- [24] Aleksander Madry. Faster approximation schemes for fractional multicommodity flow problems via dynamic graph algorithms. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 121–130. ACM, 2010.
- [25] Rajsekar Manokaran, Joseph Seffi Naor, Prasad Raghavendra, and Roy Schwartz. Sdp gaps and ugc hardness for multiway cut, 0-extension, and metric labeling. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 11–20. ACM, 2008.
- [26] Joseph Naor and Leonid Zosin. A 2-approximation algorithm for the directed multiway cut problem. *SIAM J. on Computing*, 31(2):477–482, 2001. Preliminary version in *Proc. of IEEE FOCS*, 1997.

- [27] Serge A. Plotkin, David B. Shmoys, and Éva Tardos. Fast approximation algorithm for fractional packing and covering problems. *Math. of OR*, 20:257–301, 1995. Preliminary version in Proc. of IEEE FOCS’91.
- [28] Ankit Sharma and Jan Vondrák. Multiway cut, pairwise realizable distributions, and descending thresholds. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 724–733. ACM, 2014.
- [29] Vijay V Vazirani. *Approximation algorithms*. Springer Science & Business Media, 2001.
- [30] Neal E Young. Randomized rounding without solving the linear program. In *SODA*, volume 95, pages 170–178, 1995.
- [31] Liang Zhao, Hiroshi Nagamochi, and Toshihide Ibaraki. Greedy splitting algorithms for approximating multiway partition problems. *Mathematical Programming*, 102(1):167–183, 2005.

## A Proof of Theorem 2.5

Here we prove the correctness of the integrality gap construction described in Section 2.1.

The following proposition is easy to establish based on the symmetry in the construction of the graphs.

**Proposition A.1.** *The  $s$ - $t$  cut value and the  $t$ - $s$  cut value in  $G_{i+1}$  are the same.*

Now, we calculate  $\alpha_{i+1}$  in terms of  $\alpha_i$ . We refer to the copy of  $G_i$  containing  $s$  and  $v$  with scaled capacities as  $H$ , and the one containing  $v$  and  $t$  as  $H'$ .

**Lemma A.2.** *For  $i \geq 0$ ,  $\alpha_{i+1} = \frac{4-\alpha_i}{3-\alpha_i}$ . For  $i \geq 0$ , the ratio of the one-way cut value to the two-way cut value in  $G_i$  is  $\frac{1}{\alpha_i}$ .*

**Proof:** Proof by induction on  $i$ . For the base case we see that  $\alpha_0 = 1$  and in  $G_0$  the one-way cut value and two-way cut value are both 1 and hence the ratio is equal to  $1 = \frac{1}{\alpha_0}$ .

We now prove the induction step. For this purpose we estimate the one-way cut value and the two-way cut value in  $G_{i+1}$ .

**Minimum two-way cut:** Any finite value cut that separates  $s$  from  $t$  has to cut at least one of the two edges  $(v_1, v)$ ,  $(v, v_2)$ . We consider two cases.

**Case 1:** Both  $(v_1, v)$ ,  $(v, v_2)$  are cut. To separate  $s$  and  $t$  it is best to pick a two-way cut between  $s$  and  $v$  in  $H$  (or symmetrically between  $v$  and  $t$  in  $H'$ ). Thus the total cost is  $2 + \frac{\alpha_i}{2-\alpha_i} = \frac{4-\alpha_i}{2-\alpha_i}$ .

**Case 2:** Only one of the edges  $(v_1, v)$ ,  $(v, v_2)$  is cut. Without loss of generality this edge is  $(v, v_2)$ . Since  $(v_1, v)$  is not cut  $s$  and  $t$  can reach  $v$  via  $v_1$ . Thus any two-way cut in  $G$  needs to use a one-way cut in  $H$  to separate  $v$  from  $s$  and a one-way cut in  $H'$  to separate  $v$  from  $t$ . The cost of each of these one-way cuts is, by induction,  $\frac{1}{\alpha_i} \cdot \frac{\alpha_i}{2-\alpha_i} = \frac{1}{2-\alpha_i}$ . Thus the total cost is  $1 + \frac{2}{2-\alpha_i} = \frac{4-\alpha_i}{2-\alpha_i}$ .

In both cases the cost is the same and hence the optimal two-way cut in  $G_{i+1}$  is  $\frac{4-\alpha_i}{2-\alpha_i}$ .

**Minimum one-way cut:** We now calculate one-way cut from  $s$  to  $t$ . At least one of the edges  $(v_1, v)$ ,  $(v, v_2)$  has to be cut. Also, either there is no path from  $s$  to  $v$  or no path from  $v$  to  $t$ . Thus, the cost of the one-way cut from  $s$  to  $t$  is at least  $1 + \frac{1}{2-\alpha_i} = \frac{3-\alpha_i}{2-\alpha_i}$ . Moreover it is easy to see that this is achievable by removing  $(v_1, v)$  and one-way cut from  $s$  to  $v$  in  $H$ .

**Optimum fractional solution value:** We now calculate the optimum for DIR-MC-REL on  $G_{i+1}$ . We consider the following feasible solution  $x$ . Assign 0 to the infinite weight edges and  $1/2$  to each of edges  $(v_1, v)$  and  $(v, v_2)$ . For the edges in the graphs  $H$  and  $H'$  we take an optimum solution  $y$  to DIR-MC-REL on  $G_i$  and

scale it down by  $1/2$  and assign these values to the edges of  $H$  and  $H'$ . Feasibility of  $y$  for  $G_i$  implies that distance from  $s$  to  $v$  and  $v$  to  $s$  in  $H$  according to  $x$  is  $1/2$  (since we scaled down by  $1/2$ ). It is easy to verify that distance of  $s$  to  $t$  and from  $t$  to  $s$  is 1 in the fractional solution  $x$  in  $G_{i+1}$ . Now we analyze the cost of this solution  $\sum_{e \in E(G_{i+1})} w_e x_e$ . We have a total contribution of 1 from the two edges  $(v_1, v)$  and  $(v, v_2)$ . We claim that  $\sum_{e \in E(H)} w_e x_e = \frac{1}{2} \cdot \frac{1}{\alpha_i} \cdot \frac{\alpha_i}{2-\alpha_i}$  since the cost of the two-way cut in  $H$  is chosen to be  $\frac{\alpha_i}{2-\alpha_i}$ , the integrality gap is  $\alpha_i$  and we scaled down  $y$  by  $1/2$  to obtain  $x$  in  $H$ . Same holds for  $H'$ . Thus the total fractional cost of this solution is  $1 + \frac{1}{2-\alpha_i} = \frac{3-\alpha_i}{2-\alpha_i}$ . We can see that this is an optimum solution by exhibiting a multicommodity flow of the same value for the pairs  $(s, t)$  and  $(t, s)$  in  $G_{i+1}$ . Route one unit of flow from  $s$  to  $t$  along the path  $s \rightarrow v_1 \rightarrow v \rightarrow v_2 \rightarrow t$ . In  $H$  there exists a feasible flow of total value  $\frac{1}{\alpha_i} \cdot \frac{\alpha_i}{2-\alpha_i} = \frac{1}{2-\alpha_i}$ . Let  $f(s, v)$  and  $f(v, s)$  be the amount of flow from  $s$  to  $v$  and  $v$  to  $s$  respectively. By duplicating this flow in  $H'$  we see that a flow of value  $\frac{1}{2-\alpha_i}$  exists between  $s$  and  $t$  in  $G_{i+1}$  via  $H$  and  $H'$ . Thus there is a total flow of value at least  $1 + \frac{1}{2-\alpha_i}$  in  $G_{i+1}$  and this is optimal.

We can now put together the preceding bounds to prove the lemma. The flow-cut gap in  $G_{i+1}$  is seen to be the ratio of the two-way cut value  $\frac{4-\alpha_i}{2-\alpha_i}$  and the maximum flow value  $\frac{3-\alpha_i}{2-\alpha_i}$ . Hence  $\alpha_{i+1} = \frac{4-\alpha_i}{3-\alpha_i}$  as desired. The ratio of one-way cut value  $\frac{3-\alpha_i}{2-\alpha_i}$  and the two-way cut value  $\frac{4-\alpha_i}{2-\alpha_i}$  in  $G_{i+1}$  is  $\frac{3-\alpha_i}{4-\alpha_i}$  which is equal to  $\frac{1}{\alpha_{i+1}}$ . This completes the inductive proof.  $\square$

We have a sequence of numbers  $\alpha_i$  where  $\alpha_0 = 1$  and  $\alpha_{i+1} = \frac{4-\alpha_i}{3-\alpha_i}$ . It is easy to argue that this sequence converges to 2. This proves that the integrality gap of DIR-MC-REL is in the limit equal to 2.

## B Fractionality of the LP solutions

It was shown in [19] that there is a half-integral optimum solution for the natural LP relaxation for node-weighted multiway cut (NODE-WT-MC) which was then exploited to obtain a  $2(1 - 1/k)$ -approximation. [19] also showed that the half-integral property does not hold for  $st$ -Bi-Cut. Here we generalize their example to observe that for any positive integer  $\ell$  there are examples where there may not exist an optimum solution to DIR-MC-REL on instances with two terminals that is  $1/\ell$  integral. More generally, there does not exist an edge with length more than  $1/\ell$ .

Consider the generalization of the example in [19] as shown in Fig 5. Each flow path from  $s$  to  $t$  or  $t$  to  $s$  has to use at least  $h$  edges of the type  $(u_i, u_{i+1})$  or  $(v_j, v_{j+1})$ . Since, there are only  $2(h-1)$  such edges, flow is upper bounded by  $2(h-1)/h$ . To see that this flow is also achievable, consider the following sets of paths. For  $1 \leq i \leq h-1$ , path  $P_i = s, u_1, \dots, u_{i+1}, v_i, \dots, v_h, t$  and path  $P'_i = t, v_1, \dots, v_{i+1}, u_i, \dots, u_h, s$ . Send  $1/h$  unit of flow along each of these paths. Each of the edge  $(u_j, u_{j+1})$  is part of  $P_i$  for  $i \geq j$  and part of  $P'_i$  for  $i \leq h-j$ . Hence, capacity used for edge  $(u_i, u_{i+1})$  is  $h \cdot 1/h = 1$ . Similarly for each edge  $(v_i, v_{i+1})$ . Flow value is equal to  $2(h-1)/h$ . So, optimum solution has value  $2(h-1)/h$ .

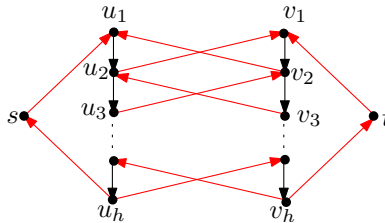


Figure 5: Edges of the form  $(u_i, u_{i+1})$  or  $(v_j, v_{j+1})$  have capacity 1 and rest have infinite capacity. Optimal fractional cut/flow is  $2(1 - 1/h)$ .

By strong duality, optimal value of DIR-MC-REL is equal to maximum flow which is equal to  $2(h-1)/h$ .

Let  $x$  be an optimal solution to the DIR-MC-REL. By feasibility of the solution, each of the paths  $P_i$  and  $P'_i$  has length at least 1. Summing up the lengths of path  $P_i$  and  $P'_i$ , we get  $\left(\sum_{j=1}^{h-1}(x(u_j, u_{j+1}) + x(v_j, v_{j+1}))\right) + x(u_i, u_{i+1}) + x(v_i, v_{i+1}) \geq 2$ . By optimality of the solution first term is equal to  $2(h-1)/h$ . Therefore,  $x(u_i, u_{i+1}) + x(v_i, v_{i+1}) \geq 2/h$ . Since, this inequality holds for all  $1 \leq i \leq h-1$ , and  $\sum_{j=1}^{h-1}(x(u_j, u_{j+1}) + x(v_j, v_{j+1})) = 2(h-1)/h$ , we get that all the inequalities are tight and  $x(u_i, u_{i+1}) + x(v_i, v_{i+1}) = 2/h$ . Since, all lengths are non-negative,  $x(u_i, u_{i+1}), x(v_i, v_{i+1}) \leq 2/h$ . By taking  $h > 2\ell$ , we get an instance where optimal solution has no edge having length at least  $1/\ell$ .